

1 We claim:

2 1. A computer program product for digitally notarizing a collection comprising a plurality of
3 data streams, the computer program product embodied on one or more computer-usable media
and comprising:

4 computer-readable program code means for computing a hash value over each of the
5 plurality of data streams, wherein each data stream is created by a different application processing
6 component;

7 computer-readable program code means for combining each hash value with a unique
8 identifier of the application processing component which created the data stream for which the
9 hash value was computed, thereby creating a combination data block;

10 computer-readable program code means for hashing the combination data block;

11 computer-readable program code means for digitally signing the hashed combination data
12 block with a private cryptographic key, wherein the private cryptographic key and a public
13 cryptographic key which is cryptographically associated therewith represent a digital notary; and

14 computer-readable program code means for providing the digitally signed hashed
15 combination data block, along with the combination data block, as the digital notarization for the
16 collection plurality of data streams, wherein the digital notarization cryptographically seals
17 contents of the collection of data streams.

1 2. The computer program product according to Claim 1, wherein:

the computer-readable program code means for combining and the computer-readable program code means for hashing operate on pairs of (hash values, identifiers), one pair for each of the plurality of data streams;

the computer-readable program code means for digitally signing digitally signs each of the

hashed pairs; and

the computer-readable program code means for providing provides the digitally signed hashed pairs, along with the hashed pairs, as the digital notarization.

3 The computer program product according to Claim 1, wherein:

the computer-readable program code means for computing a hash operates periodically, upon expiration of an elapsed time value, to compute a hash value over each of a plurality of segments of each data stream;

the computer-readable program code means for combining, the computer-readable program code means for hashing, and the computer-readable program code means for digitally signing all operate on the periodically-computed hash values for each data stream; and

8 the computer-readable program code means for providing provides the digitally signed
9 periodically-computed hash values, along with the periodically-computed hash values, as the
10 digital notarization; and

further comprising computer-readable program code means for inserting an identification of a time corresponding to each of the periodically-computed hash values at appropriate locations within each of the data streams.

1 4 The computer program product according to Claim 3, wherein the computer-readable
2 program code means for inserting uses MPEG-4 synchronization timestamping.

1 5. The computer program product according to Claim 3, wherein authenticity and integrity
2 of each of the segments is independently verifiable.

1 6. The computer program product according to Claim 3, further comprising:
2 computer-readable program code means for extracting selected ones of the segments of
3 the data streams; and

4 computer-readable program code means for verifying integrity of the extracted selected
5 ones using the public cryptographic key of the digital notary.

6 7. The computer program product according to Claim 3, further comprising:
7 computer-readable program code means for authenticating, by the digital notary, each of
8 the application processing components;
9 computer-readable program code means for extracting selected ones of the segments of
10 the data streams; and
11 computer-readable program code means for verifying authenticity of the extracted selected
12 ones using the public cryptographic key of the digital notary and the digital notarization.

13 8. The computer program product according to Claim 1, further comprising:

2 computer-readable program code means for adding an additional data stream to the
3 collection, wherein the additional data stream comprises the digital notarization.

1 9. The computer program product according to Claim 7, wherein the identifiers serve to
2 identify data streams from each of the authenticated application processing components.

1 10. The computer program product according to Claim 1, further comprising computer-
2 readable program code means for authenticating each of the application processing components
3 using the unique identifier thereof, along with a digital signature of the unique identifier that is
4 created using a private key of the application processing component.

1 11. The computer program product according to Claim 10, wherein inclusion of the unique
2 identifiers within the combination data block allows concluding that each data stream in the
3 collection was created by an authentic application processing component if operation of a
4 verification process succeeds, wherein the verification process further comprises:

5 using the public cryptographic key of the digital notary to decrypt the digitally signed
6 hashed combination data block, yielding a new version of the hashed combination data block and
7 a new version of the combination data block;
8 computing a new hash over the new version of the combination data block; and
9 determining whether the new hash is identical to the new version of the hashed
10 combination data block.

1 12. The computer program product according to Claim 11, wherein successful operation of
2 the verification process also allows concluding that the data streams in the collection have not
3 been altered.

1 13. A system for digitally notarizing a collection comprising a plurality of data streams,
2 comprising:

3 means for computing a hash value over each of the plurality of data streams, wherein each
4 data stream is created by a different application processing component;

5 means for combining each hash value with a unique identifier of the application processing
6 component which created the data stream for which the hash value was computed, thereby
7 creating a combination data block;

8 means for hashing the combination data block;

9 means for digitally signing the hashed combination data block with a private cryptographic
10 key, wherein the private cryptographic key and a public cryptographic key which is
11 cryptographically associated therewith represent a digital notary; and

12 means for providing the digitally signed hashed combination data block, along with the
13 combination data block, as the digital notarization for the collection of data streams, wherein the
14 digital notarization cryptographically seals contents of the collection of data streams.

1 14. The system according to Claim 13, wherein:

2 the means for combining and the means for hashing operate on pairs of (hash values,
3 identifiers), one pair for each of the plurality of data streams;

4 the means for digitally signing digitally signs each of the hashed pairs; and
5 the means for providing provides the digitally signed hashed pairs, along with the hashed
6 pairs, as the digital notarization.

1 15. The system according to Claim 13, wherein:
2 the means for computing a hash operates periodically, upon expiration of an elapsed time
3 value, to compute a hash value over each of a plurality of segments of each data stream;
4 the means for combining, the means for hashing, and the means for digitally signing all
5 operate on the periodically-computed hash values for each data stream; and
6 the means for providing provides the digitally signed periodically-computed hash values,
7 along with the periodically-computed hash values, as the digital notarization; and
8 further comprising means for inserting an identification of a time corresponding to each of
9 the periodically-computed hash values at appropriate locations within each of the data streams.

1 16. The system according to Claim 15, wherein the means for inserting uses MPEG-4
2 synchronization timestamping.

1 17. The system according to Claim 15, wherein integrity of each of the segments is
2 independently verifiable.

1 18. The system according to Claim 15, further comprising:
2 means for extracting selected ones of the segments of the data streams; and

3 means for verifying integrity of the extracted selected ones using the public cryptographic
4 key of the digital notary.

1 19. The system according to Claim 15, further comprising:
2 means for authenticating, by the digital notary, each of the application processing
3 components;
4 means for extracting selected ones of the segments of the data streams; and
5 means for verifying authenticity of the extracted selected ones using the public
6 cryptographic key of the digital notary and the digital notarization.

1 20. The system according to Claim 13, further comprising means for adding an additional data
2 stream to the collection, wherein the additional data stream comprises the digital notarization.

1 21. The system according to Claim 19, wherein the identifiers serve to identify data streams
2 from each of the authenticated application processing components.

1 22. The system according to Claim 13, further comprising means for authenticating each of
2 the application processing components using the unique identifier thereof, along with a digital
3 signature of the unique identifier that is created using a private key of the application processing
4 component.

1 23. The system according to Claim 22, wherein inclusion of the unique identifiers within the
2 combination data block allows concluding that each data stream in the collection was created by
3 an authentic application processing component if operation of a verification process succeeds,
4 wherein the verification process further comprises:

5 using the public cryptographic key of the digital notary to decrypt the digitally signed
6 hashed combination data block, yielding a new version of the hashed combination data block and

7 a new version of the combination data block;

8 computing a new hash over the new version of the combination data block; and

9 determining whether the new hash is identical to the new version of the hashed
10 combination data block.

11 24. The system according to Claim 23, wherein successful operation of the verification
12 process also allows concluding that the data streams in the collection have not been altered.

13 25. A method of digitally notarizing a collection comprising a plurality of data streams,
14 comprising steps of:

15 computing a hash value over each of the plurality of data streams, wherein each data
16 stream is created by a different application processing component;

17 combining each hash value with a unique identifier of the application processing
18 component which created the data stream for which the hash value was computed, thereby
19 creating a combination data block;

20 hashing the combination data block;

digitally signing the hashed combination data block with a private cryptographic key, wherein the private cryptographic key and a public cryptographic key which is cryptographically associated therewith represent a digital notary; and

providing the digitally signed hashed combination data block, along with the combination data block, as the digital notarization for the collection of data streams, wherein the digital notarization cryptographically seals contents of the collection of data streams.

26. The method according to Claim 25, wherein:

the combining step and the hashing step operate on pairs of (hash values, identifiers), one pair for each of the plurality of data streams;

the digitally signing step digitally signs each of the hashed pairs; and

the providing step provides the digitally signed hashed pairs, along with the hashed pairs, as the digital notarization.

27. The method according to Claim 25, wherein:

the step of computing a hash operates periodically, upon expiration of an elapsed time value, to compute a hash value over each of a plurality of segments of each data stream;

the combining step, the hashing step, and the digitally signing step all operate on the periodically-computed hash values for each data stream; and

the providing step provides the digitally signed periodically-computed hash values, along with the periodically-computed hash values, as the digital notarization; and

further comprising the step of inserting an identification of a time corresponding to each of the periodically-computed hash values at appropriate locations within each of the data streams.

28. The method according to Claim 27, wherein the inserting step uses MPEG-4 synchronization timestamping.

29. The method according to Claim 27, wherein integrity of each of the segments is independently verifiable.

30. The method according to Claim 27, further comprising the steps of:
extracting selected ones of the segments of the data streams; and
verifying integrity of the extracted selected ones using the public cryptographic key of the
digital notary.

31. The method according to Claim 27, further comprising the steps of:
 - authenticating, by the digital notary, each of the application processing components;
 - extracting selected ones of the segments of the data streams; and
 - verifying authenticity of the extracted selected ones using the public cryptographic key of the digital notary and the digital notarization.

1 32. The method according to Claim 25, further comprising the step of adding an additional
2 data stream to the collection, wherein the additional data stream comprises the digital
3 notarization.

1 33. The method according to Claim 31, wherein the identifiers serve to identify data streams
2 from each of the authenticated application processing components.

1 34. The method according to Claim 25, further comprising the step of authenticating each of
2 the application processing components using the unique identifier thereof, along with a digital
3 signature of the unique identifier that is created using a private key of the application processing
4 component.

1 35. The method according to Claim 34, wherein inclusion of the unique identifiers within the
2 combination data block allows concluding that each data stream in the collection was created by
3 an authentic application processing component if operation of a verification process succeeds,
4 wherein the verification process further comprises:

5 using the public cryptographic key of the digital notary to decrypt the digitally signed
6 hashed combination data block, yielding a new version of the hashed combination data block and
7 a new version of the combination data block;

8 computing a new hash over the new version of the combination data block; and
9 determining whether the new hash is identical to the new version of the hashed
10 combination data block.

1 36. The method according to **Claim 35**, wherein successful operation of the verification
2 process also allows concluding that the data streams in the collection have not been altered.

1 37. A digitally notarized collection of data streams, comprising:
2 a plurality of data streams in the collection, wherein each data stream is created by a
3 different application processing component; and
4 a digital notarization of the collection, created by the steps of:
5 computing a hash value over each of each of the plurality of data streams;
6 combining each hash value with a unique identifier of the application processing
7 component which created the data stream for which the hash value was computed, thereby
8 creating a combination data block;
9 hashing the combination data block;
10 digitally signing the hashed combination data block with a private cryptographic
11 key, wherein the private cryptographic key and a public cryptographic key which is
12 cryptographically associated therewith represent a digital notary; and
13 providing the digitally signed hashed combination data block, along with the
14 combination data block, as the digital notarization for the collection of data streams, wherein the
15 digital notarization cryptographically seals contents of the collection of data streams.

1 38. A method of doing business using digitally notarized data streams, comprising steps of:

2 digitally notarizing a collection comprising a plurality of data streams, further comprising
3 steps of:

4 computing a hash value over each of the plurality of data streams, wherein each
5 data stream is created by a different application processing component;
6 combining each hash value with a unique identifier of the application processing
7 component which created the data stream for which the hash value was computed, thereby
8 creating a combination data block;

9 hashing the combination data block;
10 digitally signing the hashed combination data block with a private cryptographic
11 key, wherein the private cryptographic key and a public cryptographic key which is
12 cryptographically associated therewith represent a digital notary; and

13 providing the digitally signed hashed combination data block, along with the
14 combination data block, as the digital notarization for the collection of data streams; and

15 verifying authenticity of the digitally notarized collection of data streams, by a receiver of
16 the digital notarization, further comprising:

17 using the public cryptographic key of the digital notary to decrypt the digitally
18 signed hashed combination data block, yielding a new version of the hashed combination data
19 block and a new version of the combination data block;

20 computing a new hash over the new version of the combination data block; and
21 determining whether the new hash is identical to the new version of the hashed
22 combination data block, and if so, concluding that the data streams in the collection have not been
23 altered.